DEVICE TESTING CONTROL

Background of the Invention

[0001] Prior to shipping a device, such as a system-on-a-chip (SOC), the device must be tested to determine whether it has been manufactured correctly and is fully operational. A variety of testers are available for such testing. Typically, a tester is a very large and expensive machine which is designed to precisely position the placement of logic signal transitions at very high speeds. Most testers are aimed at creating a "functional environment" for a device which mimics the environment in which the device will eventually be used, to thereby demonstrate that the device will behave as expected in that environment.

[0002] For functional tests, a tester applies a series of "test vectors" to the inputs of the device. A test vector is a critically timed cycle of events lasting a short period of time referred to as a "vector cycle". Within a vector cycle, and at precisely calculated times, logic signal drivers in the tester apply stimulus to device inputs. At the same or some precisely delayed time, logic signal comparators in the tester monitor responses at device outputs. When many test vectors are executed sequentially, discrepancies between monitored and expected device outputs, if any, are noted as device failures.

[0003] An alternative or adjunct to functional testing is "structural" testing (e.g., "scan" testing). Structural testing enables the testing of structures which are deeply embedded within a device. Rather than testing

the device's internal structure by applying stimulus to the device's inputs, structural testing involves shifting a series of test vectors into the core of a device, and after each test vector is shifted in, launching the test vector and capturing a response. Each response is then shifted out of the device. In this manner, a tester can verify that all of a device's elements are present and operational. An assumption of structural testing is that if all elements are present and operational, then the elements will contribute to performing the greater and intended functions of the device (e.g., adding, shifting, etc.), and the device will function as designed.

[0004] During testing, hundreds of devices having the same design may need to be tested. Additionally, each device under test may be subjected to multiple tests of varying types. In some cases, the selection of tests to be executed may vary depending upon the results of previous tests.

Summary of the Invention

[0005] Methods and systems for controlling device testing are disclosed. In one embodiment, the system comprises a local controller having a slave mode and a control mode. When in the control mode, the local controller is to control testing of a device and to initiate one or more test instructions to be applied to the device. When in the slave mode, the local controller passes through a remote test instruction received from a remote controller to a tester. The system further comprises the tester, communicatively coupled to the local controller, to apply one of the one or more test instructions and the remote test instruction to the device.

Brief Description of the Drawings

[0006] Illustrative embodiments of the invention are illustrated in the drawings in which:

[0007] FIG. 1 illustrates a first exemplary configuration of a system, which may be used to test one or more devices;

[0008] FIG. 2 illustrates a second exemplary configuration of a system, which may be used to test one or more devices;

[0009] FIG. 3 illustrates an exemplary method that may be used by the local controller of FIGS. 1 or 2 to switch between modes of test control;

[0010] FIG. 4 illustrates an exemplary method that may be used by the configuration of FIG. 2 during device testing; and

[0011] FIG. 5 illustrates an exemplary method that may be used by the configuration of FIG. 1 during device testing.

Detailed Description

[0012] An exemplary embodiment of a system that may be used to test one or more devices is illustrated in FIG. 1. The system includes a local controller 160 having at least two modes of operation: a control mode, and a slave mode. When in the control mode, local controller 160 controls testing of device 150. The control of testing may include initiation of one or more test instructions to be applied to device 150. By way of example, the test instructions initiated by local controller 160 may be selected from tests in a test file or another location. Depending upon factors, such as the results of one or more previous executions of test instructions and instructions found in a test file, local controller 160 may control branching, iteration, and looping of the tests in the test file and initiate test instructions to tester 100 accordingly.

[0013] Local controller 160 may also control the application of test instructions to multiple devices while in the control mode. For example, local controller 160 may communicate with a prober/handler (not shown) to direct the removal of device 150 after testing is complete and the insertion of another device to be tested. The second device may have the same configuration as device 150. Thus, local controller 160 may manage testing of multiple lots, cassettes, or other groupings of devices. Alternately, the second device may have a different configuration with different tests that need to be applied.

[0014] In the control mode, local controller 160 may also manage test results received from the tester 100 as a result of applying the one or more test instructions to device 150. As previously described, test results may be used by local controller 160 to determine which test instruction should be sent

to tester 100. Additionally, in some embodiments, management of the test results may include compiling a plurality of test results from multiple test instructions applied to device 150. If multiple devices of the same configuration were tested, local controller may also compile the results for the lots, cassettes, or other grouping of devices. In some embodiments, the individual or compiled results may be used by local controller, or other component, to determine if a device, lot, cassette, or other grouping passed device testing.

[0015] As shown in FIG. 2, in a second exemplary configuration, local controller 160 may be communicatively coupled to a remote controller 200. By way of example, remote controller 200 may be a station controller as defined by the SEMI (Semiconductor Equipment and Materials International) Station Controller Architecture. In this configuration, remote controller 200 controls the testing of device 150 and local controller operates in slave mode. In the slave mode, local controller 160 may pass through test instructions received from the remote controller 200 to the tester 100. Local controller 160, in the slave mode, may also pass through results of test instructions to the remote controller 200. The results passed through to the remote controller 200 may also be compiled by local controller 160 for local reference.

[0016] In some embodiments, local controller 160 may be communicatively coupled to memory (not shown) to receive test instructions from the remote controller 200. Local controller 160 may poll the memory to detect the presence of a test instruction initiated by remote controller 200. If local controller 160 is in control mode and an instruction from the remote controller 200 is detected, local controller 160 may switch from the control

mode to the slave mode. In alternate embodiments, a flag or semaphore may be used to switch local controller 160 between its two modes. Alternately, a parameter may be configured to direct local controller 160 to operate in either the slave mode or control mode. It should be appreciated that alternate methods may also be used to switch modes. It should also be appreciated that local controller 160 may be software, firmware, hardware, or a combination of these and in some embodiments may be a part of tester 100.

[0017] In the configurations of both FIGS. 1 and 2, local controller 160 is communicatively coupled to tester 100. By way of example, tester 100 may be a system-on-a-chip (SOC) tester. Tester 100 may include a plurality of boards 102-106. Each board may include a plurality of pins (not shown) to apply logic signals to a device 150 under test according to test instructions received from local controller 160. The pins may also be used to receive outputs from device 150. In one embodiment, each pin may include its own memory to use during the testing of the device. The memory may be used to store pin specific vector information. In alternate embodiments, memory may not be included on each pin, but may instead be included for each board or other component of tester 100. In alternate embodiments other types and configurations of testers may also be used to apply test instructions to device 150.

[0018] In one embodiment, the device 150 under test may be a system-on-a-chip (SOC). In alternate embodiments, device 150 may be a wafer having multiple devices to be tested or other type of circuit device. Additionally, at times between testing of devices, tester 100 will not be coupled to a device 150.

[0019] It should be appreciated that tester 100 and local controller 160 may be used in configurations in which local control of testing is needed and configurations in which a remote controller 200 is used to control testing. This may allow a supplier to sell the same tester 100 and local controller 160 in a variety of customer configurations. Additionally, customers may change between local and remote controlled configurations without needing to change tester 100 or local controller 160.

[0020] FIG. 3 illustrates an exemplary method that may be used by local controller 160 to switch between modes of test control. First, a test instruction received from a remote controller 200 is detected 300. By way of example, the test instruction may be detected 300 by polling a memory or checking a semaphore. After the test instruction is detected 300, local controller 160 switches 305 from control mode to slave mode.

[0021] After local controller 160 switches 305 to slave mode, the test instruction may be passed through to tester 100 and the tester 100 may then apply the test instruction to a device 150. Local controller 160 may then pass through the test result to remote controller 200. In some embodiments, local controller 160 may compile the result with one or more additional results that were previously passed through to the remote controller 200 so that this information may be locally available.

[0022] It should be appreciated that in alternate embodiments, other methods may be used to switch local controller between modes of test control. By way of example, a configurable parameter may be provided to direct local controller 160 to operate in slave mode or control mode.

Alternately, local controller 160 may initially operate in slave mode until an event happens to trigger local controller 160 to switch to control mode.

[0023] A simplified explanation of the differences between testing control when local controller 160 is in slave mode and when local controller 160 is in control mode can be given with reference to FIGS. 4 and 5. FIG. 4 illustrates an exemplary flow of testing that may be used while local controller 160 is in slave mode. In the slave mode, a test instruction from remote controller is received 400 by local controller 160. The test instruction is then passed through 405 to tester 100.

[0024] After the test instruction has been applied to device 150 by tester 100, a result of the test instruction is passed through 410 to remote controller 200. This process may be repeated for additional test instructions. Thus, in slave mode, the testing of device 150, including the selection of test instructions to be applied, is done by remote controller 200.

[0025] In contrast, as illustrated in FIG. 5, while in control mode, local controller 160 controls the testing of device 150. Local controller 160 sends 505 a test instruction to tester 100. As previously described, local controller 160 may obtain the test instructions to be sent to tester 100 from tests in a test file or another location.

[0026] After the tester 100 has applied the test instruction to device 150, local controller 160 receives 510 the test result. Local controller 160 then determines 515 if there are more tests that need to be applied to the device 150 currently being tested. If there are more tests, local controller 160 sends 505 the next instruction to tester 100. During testing, local controller 160 may branch, iterate, and loop through tests found in a test file and may

examine test results to determine which test instruction should be sent to tester 100.

[0027] If 515 there are no more tests for device 150, local controller 160 may determine 520 if there are more devices with the same configuration as device 150 that need to be tested. If not, testing ends 530. If there are more devices, local controller 160 may direct a prober/handler (or other mechanism) to change the device 525 to the next device that needs to be tested. A test instruction to be applied to the new device is then sent 505 to the tester 100. Thus, control of the device testing is done by local controller 160 when local controller 160 is in control mode.

[0028] While illustrative and presently preferred embodiments of the invention have been described in detail herein, it is to be understood that the inventive concepts may be otherwise variously embodied and employed, and that the appended claims are intended to be construed to include such variations, except as limited by the prior art.